PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re the Application of

Chang et al. (SANDP039)	Conf. No. 8920
Serial No. 10/679,000	Group Art Unit: 2186
Filed: October 2, 2003	Examiner: Tsai

For: Method and Apparatus for Managing the Integrity of Data in Non-volatile Memory System

APPELLANTS' BRIEF

Commissioner for Patents
Washington, DC 20231

Dear Sir:

Appellants respectfully present their brief in support of their appeal of the final rejection of claims in this case. The Notice of Appeal was filed on July 4, 2007, as indicated on the date of the automated receipt from the Patent and Trademark Office.

Real Party in Interest

The real party in interest in this application is SanDisk Corporation.

Related Appeals and Interferences

Copending application S.N. 10/678,893 is currently on appeal, and contains claims relative to which claims in this case have been finally rejected under the judicially-created doctrine of double patenting of the obviousness type.

The undersigned is aware of no other related applications that are currently on appeal or in an interference that would be directly affected by, or that themselves directly affect or have a bearing on, this appeal.

Status of the Claims

Claims 1 through 25 and 27 through 31 were finally rejected in the Office Action of April 4, 2007, and are the subject of the present appeal.

Claim 26 has been canceled.

Status of Amendments

No amendment was presented after the final rejection.

Summary of the Claimed Subject Matter

Independent claim 1 is directed to a method of error-correction coding of data stored in a non-volatile memory system (100), such as a flash memory system.[1] According to the claimed method, a page (400) of data to be written into non-volatile memory (124) is divided into two or more segments of data (406, 408; 426a, 426b, 426c; 446a, 446b, 444).[2] One of the segments (406; 426a; 446a) within the page is encoded according to a first error-correction code (ECC) algorithm, and another segment (408; 426b; 446b) within the page is separately encoded according to a second ECC algorithm.[3]

Independent claim 11 is directed to a memory system (100) including a non-volatile memory, and also including code devices (716) stored in a memory area (128) of the system.[4] These code devices (716) include code devices that divide a page of the non-volatile memory

[1] Specification of S.N. 10/679,000, *as published as* U.S. Patent Application Publication No. US 2004/0083334 A1, published April 29, 2004, paragraphs [0035] through [0048]; Figures 3a through 3c.
[2] Specification, *supra,* paragraphs [0049] through [0057]; Figures 4a through 4c.
[3] Specification, *supra,* paragraph [0071].
[4] Specification, *supra,* paragraphs [0035] through [0048]; [0063]; [0068].

into at least two segments,[5] and code devices that encode a first one of the segments according to a first ECC algorithm, and that encode a second segment according to a second ECC algorithm.[6]

Independent claim 23 is also directed to a memory system (100) including a non-volatile memory (124).[7] The memory system of claim 23 includes means (128; 716) that divide at least part of a page of the non-volatile memory into two or more segments, and means that perform error correction code calculations on a first segment according to a first ECC algorithm, and separately upon a second segment according to a second ECC algorithm.[8]

The claimed invention, in both its method and system form, provides important advantages over conventional flash memory systems. As known in the art, error correction coding impacts the data storage capacity, because such coding requires some number of bits to store redundant information used to detect and correct errors in the payload data. Higher capability error correction codes (*i.e.,* codes that correct a higher number of errors for a given data block size) require more computational overhead and also more memory bits to store the redundant data, as compared with lower capability codes.[9] The claimed method and systems, and their ability to use different ECC codes for different portions of the same page, provide great flexibility in the optimizing of error correction performance, for example in the storage of information of differing sensitivity to error, and improved efficiency in using more complex ECC operations only to the extent necessary for the desired error correction capability.

Grounds of Rejection to Be Reviewed On Appeal

The double patenting rejection

Claims 1, 3, 4, 6, 7, 10, 11, 15, 21, 23, 27, and 28 were provisionally rejected under the judicially-created doctrine of double patenting of the obviousness type, relative to claims 4, 3, 6, 9, and 10 of copending application S.N. 10/678,893, as amended on December 19, 2006, and now on appeal.

---

[5] Specification, *supra,* paragraphs [0049] through [0057]; Figures 4a through 4c.
[6] Specification, *supra,* paragraph [0071].
[7] Specification, *supra,* paragraphs [0035] through [0048];
[8] Specification, *supra,* paragraphs [0035] through [0057]; [0063]; [0068]; [0071].
[9] Specification, *supra,* paragraphs [0010]; [0032].

The §103 rejection of claim 1 and its dependent claims

Independent method claim 1 and its dependent claims 2, 3, and 6 through 9, were finally rejected, under §103, as unpatentable over the Bassett et al. reference[10] in view of the Katayama et al. reference[11]. The Examiner asserted that the Bassett et al. reference taught all of the steps of the method of claim 1, but in connection with a disk drive memory, rather than a flash memory as claimed.[12] The Examiner found, however, that the Katayama et al. reference teaches the applying of different ECC algorithms to different blocks of a flash memory disk, such that it would have been obvious for one skilled in the art to recognize that the method taught by Bassett et al. would be applicable to a flash memory system, as demonstrated by Katayama et al.[13] The claims were rejected accordingly.

Claims 4 and 5 were rejected under §103 as unpatentable over the Bassett et al. and Katayama et al. references, as applied against claim 1, and further in view of an alleged admission by Applicants and the Zhang et al. reference[14], in connection with the particular features of the error correction codes recited in these claims. Claim 10 was finally rejected under §103 as unpatentable over the Bassett et al. and Katayama et al. references, as applied against claim 1, and further in view of the Kramer reference[15], in connection with the NAND and MLC NAND flash memory limitation of the claim.

The §103 rejection of claim 11 and its dependent claims

Claim 11 and its dependent claims 12 through 14, 17 through 20, and 22 were finally rejected, under §103, as unpatentable over the Bassett et al. reference in view of the Katayama et al. reference, in the manner discussed above relative to claim 1.[16] Claims 15 and 16 were finally rejected under §103 as unpatentable over the et al. and Katayama et al. references, as applied against claim 11, and further in view of the alleged admission by Applicants and the Zhang et al. reference, on a similar basis as asserted against claims 4 and 5. Claim 21 was finally rejected

[10] U.S. Patent No. 6,747,827 B1, issued June 8, 2004 to Bassett et al.
[11] U.S. Patent No. 6,651,212 B1, issued November 18, 2003 to Katayama et al.
[12] Office Action of April 4, 2007, page 11.
[13] Office Action, *supra*, page 12.
[14] U.S. Patent No. 6,662,333, issued December 9, 2003 to Zhang et al., on an application filed February 4, 2000.
[15] U.S. Patent No. 6,182,239, issued January 30, 2001 to Kramer.
[16] Office Action, *supra*, page 15.

under §103 as unpatentable over the Bassett et al. and Katayama et al. references, as applied against claim 11, and further in view of the Kramer reference, as discussed above relative to claim 10.

The §103 rejection of claim 23 and its dependent claims

Claims 23 and its dependent claims 24, 25, and 27 through 30 were finally rejected, under §103, as unpatentable over the Bassett et al. reference in view of the Katayama et al. reference, in the manner discussed above relative to claims 1 and 11. Claim 31 was rejected under §103 as unpatentable over the Bassett et al. and Katayama et al. references, as applied against claim 23, and further in view of the Kramer reference, as discussed above relative to claims 10 and 21.

Argument

The double patenting rejection

Appellants submit that the provisional double patenting rejection of claims 1, 3, 4, 6, 7,10, 11, 15, 21, 23, 27, and 28 is in error, and should be reversed, because the Examiner erred in his determination that these claims are obvious variations of those in copending application S.N. S.N. 10/678,893.

The asserted claims of copending application S.N. 10/678,893 recite the writing of data encoded using a first error detection algorithm into a first *block* of a non-volatile memory, and the writing of data encoded using a second error detection algorithm into a second *block* of the non-volatile memory. In contrast, the claims in this case rejected under the double patenting doctrine recite the encoding of data associated with a first *segment* of a *page* of a non-volatile memory according to a first error correction code (ECC) algorithm, and the encoding of data associated with a second *segment* of a *page* of that memory according to a second ECC algorithm.[17] In making the rejection, the Examiner asserted that the claims of copending application S.N. 10/678,893 are narrower than the rejected claims in this case, and further that:

[17] *See, e.g.,* claim 1 on appeal in this case, and claim 4 in copending application S.N. 10/678,893.

> . . . every non-volatile flash memory inherently has a smallest unit of programming, which may be referred to by various terms, such as page, sector, segment, block, or whatever.[18]

Appellants submit that the determination that the segment and page of a non-volatile memory recited in the claims in this application are different from the blocks of copending application S.N. 10/678,893, and that therefore the Examiner is incorrect in determining that the terms are interchangeable, and is incorrect in determining that the claims in the copending application S.N. 10/678,893 are narrower than the claims in this case.

The claims of a patent or patent application are to be interpreted from the claim language itself, to consider its ordinary and customary meaning to a skilled artisan at the time of invention, in the context of the entire patent including the specification.[19] Reliance on the context of the invention is appropriate because patent claims do not stand alone, but are part of a fully integrated written instrument.[20]

Accordingly, to determine the meaning of "segment", "page", and "block" in the claims of this application, one must consider the context of the entire patent including the specification. As evident from the specification of this application, these terms refer to particular portions of a non-volatile flash memory.[21] More specifically, the specification of this application refers to a "block", in the manner "common for flash EEPROM systems", as "typically the smallest unit of erase", "[t]hat is, each block contains the minimum number of memory cells that are erased together."[22] In addition, this specification states that, in the context of this invention, a "block" is "divided into a number of pages".[23] Regarding a "page" of the flash memory, the specification of this application states that "those skilled in the art" will appreciate that "a page may be the smallest unit of programming," "[t]hat is, a basic programming operation writes data into or reads data from a minimum of one page of memory cells".[24] And, for purposes of the claims on

[18] Office Action, *supra*, page 8.
[19] *Phillips v. AWH Corp.*, 415 F.3d 1303, 75 USPQ2d 1321 (Fed. Cir. 2004); *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 62 USPQ2d 1658 (Fed. Cir. 2002); *Johnson Worldwide Associates, Inc. v. Zebco Corp.*, 175 F.3d 985, 50 USPQ2d 1607 (Fed. Cir. 1999).
[20] *Phillips v. AWH Corp., supra*.
[21] Specification, *supra,* paragraphs [0006] through [0008]; [0044]; [0048].
[22] Specification, *supra,* at paragraph [0044].
[23] *Id.*
[24] *Id.*

appeal in this application, the specification refers to such a "page" as being "divided into segments which may be separately encoded using an ECC algorithm".[25] Given this description, which clearly refers to blocks as comprised of a number of pages, and pages as divided into segments, the Examiner's interpretation of the smallest unit of programming of a flash memory as being referred to "by various terms, such as page, sector, segment, block, or whatever"[26], is clearly in error if applied to the claims in this application.

Appellants also submit that the proper interpretation of the claim term "block" in copending application S.N. 10/678,893 is the same as in this application. In this regard, the specification of copending application S.N. 10/678,893 includes the identical description of a "block" and of a "page" in its flash memory as does the specification in this application, to wit:

> As is common for flash EEPROM systems, the block is typically the smallest unit of erase. That is, each block contains the minimum number of memory cells that are erased together. Each block is typically divided into a number of pages. As will be appreciated by those skilled in the art, a page may be the smallest unit of programming. That is, a basic programming operation writes data into or reads data from a minimum of one page of memory cells.[27]

Therefore, the term "block" has the same meaning both in this application and also in copending application S.N. 10/678,893, and the term "page" has the same meaning both in this application and also in copending application S.N. 10/678,893. And in both this application and also in copending application S.N. 10/678,893, there is no reasonable interpretation under which the terms "block" and "page" mean the same thing, much less under which "block" and "segment" (a segment being part of a page as discussed above) mean the same thing.

Appellants therefore submit that the determination by the Examiner that the claims in copending application S.N. 10/678,893 are narrower than the claims in this case is in error. The claims in copending application S.N. 10/678,893 recite, as mentioned above, the writing of data encoded using a first error detection algorithm into a first *block* of a non-volatile memory, and the writing of data encoded using a second error detection algorithm into a second *block* of the

[25] Specification, *supra,* at paragraph [0048].
[26] Office Action, *supra,* page 8.
[27] Application S.N. 10/678,893, *as published as* U.S. Patent Application Publication No. US 2004/0083333, published April 29, 2004, paragraph [0046].

non-volatile memory. In contrast, the claims in this case recite the encoding of data associated with a first *segment* of a *page* of a non-volatile memory according to a first error correction code (ECC) algorithm, and the encoding of data associated with a second *segment* of a *page* of that memory according to a second ECC algorithm. One can therefore practice the invention claimed in copending application S.N. 10/678,893 without practicing the invention of the claims in this case, and one can practice the invention of the claims in this case without necessarily practicing the invention claimed in copending application S.N. 10/678,893. The scope of the claims in these applications is simply different. To the extent that the double patenting rejection is based on this determination that the claims in copending application S.N. 10/678,893 are narrower than claims on appeal in this case, Appellants submit that the rejection is in error.

Furthermore, Appellants submit that the basis, by which the Examiner asserts that the claims in this case are an obvious variation from those in copending application S.N. 10/678,893, is in error. The Examiner's assertion that "every non-volatile flash memory inherently has a smallest unit of programming, which may be referred to by various terms, such as page, sector, segment, block, or whatever"[28] is clearly contrary to the specifications of the applications at issue, and is thus clearly contrary to any reasonable interpretation of the claims in those applications. The conclusion that these claims are an obvious variation of the claims in copending application S.N. 10/678,893 is therefore in error.

For these reasons, Appellants submit that the double patenting rejection of claims 1, 3, 4, 6, 7,10, 11, 15, 21, 23, 27, and 28 is in error, and should be reversed.

The §103 prior art rejections

A *prima facie* obviousness determination of patent claims requires the articulation of reasoning, with some rational underpinning, supporting the conclusion that one skilled in the art would have combined or modified the known elements to reach the claimed subject matter.[29] If the Examiner fails to establish such a *prima facie* case, the obviousness rejection is improper and

[28] Office Action, *supra*, page 8.
[29] *KSR International Co. v. Teleflex Inc et al.*, 550 U.S. _____; 127 SCt 1727; 167 L.Ed.2d 705; 82 USPQ2d 1385 (2007)..

should be overturned on appeal.[30] In such a determination based on the combination of prior art references, there must be some suggestion or motivation to combine these references, beyond a mere conclusory statement, to avoid a conclusion that the combination is simply an improper use of the inventor's own teachings in hindsight.[31]

Claim 1 and its dependent claims

Appellants respectfully submit that the Examiner has failed to establish a *prima facie* case of obviousness relative to claim 1, because there is no valid reasoning articulated by the Examiner to apply the teachings of the Bassett et al. reference to a flash memory, and thus in such a manner as to reach independent claim 1. More specifically, the teachings of the Katayama et al. reference asserted by the Examiner as providing such suggestion arise from a misinterpretation of the Katayama et al. reference. Appellants therefore submit that the final rejection of claim 1 and its dependent claims is in error and should be reversed.

As mentioned above, the Examiner asserted that the Bassett et al. reference taught the steps of the method of claim 1, but in connection with a disk drive memory; the rejection is based on the Examiner's assertion that the Katayama et al. reference teaches the applying of different ECC algorithms to different blocks of a flash memory disk.[32] The specific language of the rejection is instructive:

> Further, Katayama et al. disclose in their invention . . . a method of applying different ECC algorithms to different blocks of a flash memory disk ....
>
> It is further noted that both Bassett et al. and Katayama et al. teach that the motivation of using different ECC algorithms to different blocks of memory instead of uniformly applying the same ECC algorithm to all blocks of memory....
>
> Therefore, it would have been obvious for one of ordinary skills in the art at the time of Applicants' invention to recognize that the method disclosed by Bassett is equally applicable to a flash memory, as demonstrated by Katayama, and to incorporate the method disclosed by Bassett to a flash memory system as flash

[30] *In re Rijckaert*, 9 F.3d 1531, 1532, 28 USPQ2d 1955, 1956 (Fed. Cir. 1993).
[31] *KSR, supra; In re Dembiczak*, 175 F.3d 994, 999, 50 USPQ3d 1614 (Fed. Cir. 1999) ("Combining prior art references without evidence of such a suggestion, teaching, or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability -- the essence of hindsight.").
[32] Office Action, *supra,* pages 11 and 12.

> memory becomes more commonly used in the industries, as demonstrated by Katayama.[33]

Appellants submit, however, that the Katayama et al. reference does not teach what the Examiner says it teaches. And because it does not teach what the Examiner says it teaches, the Katayama et al. reference fails to provide the suggestion that the Examiner uses to reject claim 1 and its dependent claims.

The Katayama et al. reference does not teach the applying of different ECC algorithms to different blocks of a flash memory. Rather, it teaches the applying of two different ECC algorithms sequentially to the same data. For example, the Katayama et al. reference teaches:

> An error correction code check symbol consists of a 5-symbol outer code check symbol C1 which is put to data by the ECC circuit 107 in the controller 102, and a 3-symbol inner check symbol C2 which is put to the data by the error correction coding circuit 609 in the on-chip ECC circuit 120 on the flash memory chip 111.[34]

The reference provides a detailed description of the applying of the "outer" code check symbol C1 by the ECC circuit 107:

> Next, the data write operation will be explained in detail.
>
> Data of the 512 bytes by 8 bits unit size is put in the flash memory disk 101 from an external device over the external bus 109. The controller 102 of the flash memory disk 101 transfers the input data to the ECC circuit 107 through the external bus interface 106.
>
> The ECC circuit 107, which uses the Galois field of GF (2.sup.10) appends 2 bit's dummy data of "0" to each byte of the 512 bytes by 8 bits data, i.e., d511 (511th byte), d510 (510th byte), d509, . . . , d0 (d511 through d0 are expressed by 8 bits of GF (2.sup.8), to produce 512 bytes by 10 bits data, i.e., d511', d510', d509', . . . , d0', thereby to get a code polynomial of formula (4).
>
> . . . . Next, from the polynomial (6) , R14, R13, R12, R11 and R10 are obtained as the outer code check symbol C1. [35]

Following the generation of the "outer" code check symbol C1 by the ECC circuit 107, the operation taught by the Katayama et al. reference then generates the "inner" check symbol C2 by operation of the error correction coding circuit 609. This "inner" check symbol C2 is clearly

---

[33] *Id.* (emphasis in original).
[34] Katayama et al., *supra,* column 10, lines 51 through 56.
[35] *Id.,* at column 11, lines 19 through 45.

taught by the reference as generated from both the original data and also the "outer" check symbol C1 produced by ECC circuit 107:

> The controller 102 next transfers the input data of 512 bytes by 8 bits and the outer code check symbol C1 of 5 symbols by 10 bits to one of the flash memory chips 111,112,113 or 114 in the order of d511, d510, . . . , d0, R14, R13, R12, R11 and R10 through the internal bus interface 105. Transfer to the flash memory chip 111 is assumed here.
>
> The error correction coding circuit 609 in the on-chip ECC circuit 120 of the flash memory chip 111, which uses the Galois field of GF(2.sup.10), appends 2 bit's dummy data of "0" to each byte of the 512 bytes by 8 bits data [d511,d510,d509, . . . ,d0] to produce 512 bytes by 10 bits d [d511',d510',d509', . . . ,d0'], and adds the outer code check symbol C1 symbols by 10 bits [R14, R13, R12, R11 and R10] thereby to get a codeepolynomial [sic] of formula (7).
>
> . . . .
>
> $$d511'x^{519} + d510'x^{518} + \ldots + d1'x^{9} + d0'x^{8} + R14x^{7} + R13x^{6} + R12x^{5} + R11x^{4} + R10x^{3}) \bmod (x+\alpha^{5})(x+\alpha^{6})(x+\alpha^{7}) = R22x^{2} + R21x + R20 \quad (9)$$
>
> Next, from the polynomial (9), R22, R21 and R20 are obtained as the inner code check symbol C2.[36]

As evident from this description in the Katayama et al. reference itself, the "outer" code check symbol C1, consisting of the values R14, R13, R12, R11 and R10, is included in the data block (along with data byte values d511', d510', ... d0')[37] from which the inner code check symbol C2, consisting of the values R22, R21, R20, is produced by error correction coding circuit 609. The ultimate encoded data that is written to each, and uniformly every, flash memory block of the Katayama et al. reference is therefore encoded in the same manner, namely sequentially by first generating the "outer" code check symbol C1 from the original data, and then generating the "inner" code check symbol C2 from the original data and the "outer" code check symbol C1.

This interpretation of the operation of the Katayama et al. reference is not inconsistent with the portions of the reference cited by the Examiner in making the rejection, namely its abstract:[38]

---

[36] *Id.*, column 11, line 46 through column 12, line 7.
[37] *Id.*, column 11, lines 24 through 30.
[38] Office Action, *supra,* page 12.

> An ECC circuit implements a first error correction using a first BCH connection code and flash memory chips implement a second error connection using a second BCH error correction code which uses the same Galois filed. A controller implements error detection based on the first error correction code and using the information provided by the flash memory chip. . . .[39]

There is no mention in this passage of the abstract, nor elsewhere in the abstract, nor elsewhere in the Katayama et al. reference, that different ECC algorithms are applied to different blocks of the flash memory of Katayama et al.[40]

Therefore, Appellants submit that the Katayama et al. reference does not teach what the Examiner asserted it teaches in making the final rejection. Specifically, the Katayama et al. reference does not in fact teach "a method of applying different ECC algorithms to different blocks of a flash memory disk", as asserted by the Examiner.[41] And therefore, the teachings of the Katayama et al. reference do not "teach that the motivation of using different ECC algorithms to different blocks is to save memory space", nor do the teachings of the Katayama et al. reference "demonstrate" that the method disclosed by Bassett et al. is equally applicable to a flash memory.[42]

This alleged, and erroneous, basis is the only basis asserted by the Examiner for applying the teachings of the Bassett et al. reference in a flash memory, in such a manner as to reach claim 1 and its dependent claims. But because this alleged basis is in error, as discussed above, Appellants therefore submit that the Examiner's contention that it would have been obvious to apply the teachings of the Bassett et al. reference to a flash memory is not based on an articulation of valid reasoning, with at least some rational and valid underpinning, as is required to support a §103 rejection.[43] Appellants therefore submit that the final rejection of claim 1 and its dependent claims 2 through 10 is in error because the Examiner has failed to present a *prima facie* determination of obviousness of those claims. Reversal of the final rejection is requested.

---

[39] Katayama et al., *supra,* abstract.
[40] Nor do the other references applied against others of the claims, including the Zhang et al. reference, the Kramer reference, and Appellants' alleged admissions, provide such "demonstration", nor were they asserted as doing so.
[41] Office Action, *supra,* page 11.
[42] Office Action, *supra,* page 12.
[43] *KSR, supra.*

Claim 11 and its dependent claims

Appellants respectfully submit that the Examiner has failed to establish a valid *prima facie* case of obviousness relative to claim 11 and its dependent claims, because the sole reasoning articulated by the Examiner to apply the teachings of the Bassett et al. reference to reach independent claim 11 is in error. In particular, Appellants submit that the Examiner misinterpreted the Katayama et al. reference, and submit that the rejection of claim 11 and its dependent claims is founded on this misinterpretation and is not otherwise supported. Appellants therefore submit that the final rejection of claim 11 and its dependent claims is in error and should be reversed.

As discussed above relative to claim 1, the Examiner asserted that the Bassett et al. reference taught the elements of the claim in connection with a disk drive memory, and that the Katayama et al. reference demonstrates that the Bassett et al. teachings can be also applied to a flash memory, thus reaching claim 11. The Examiner based this conclusion on his finding that the Katayama et al. reference teaches the applying of different ECC algorithms to different blocks of a flash memory disk. [44] However, as discussed above relative to claim 1, Appellants submit that the Katayama et al. reference does not, in fact, teach applying different ECC algorithms to different blocks of flash memory.

While the Katayama et al. reference teaches the applying of two different ECC polynomials, these two polynomials are applied to the same block of data in sequence, and not to different flash memory blocks. As discussed above in detail, the Katayama et al. reference teaches the encoding of a data word by way of an ECC circuit 107 deriving and then applying an "outer" code check symbol C1 to the data word. [45] Once the "outer" code check symbol C1 is derived, the reference then teaches that a data block including the original data and this "outer" code check symbol C1 is encoded by an error correction coding circuit 609 to generate an "inner" check symbol C2. [46] The data written to each and every flash memory block of the Katayama et al. reference is encoded in this same manner. The reference discloses no circuitry

[44] Office Action, *supra*, pages 11 and 12 ("Further, Katayama et al. disclose in their invention . . . a method of applying different ECC algorithms to different blocks of a flash memory disk."

[45] *Id.*, at column 11, lines 19 through 45.

[46] *Id.*, column 11, line 46 through column 12, line 7.

or code devices that writes data to one flash memory block according to one ECC algorithm and that writes data to another flash memory block according to another ECC algorithm. Rather, each flash memory block of the Katayama et al. reference receives data encoded according to the same two algorithms, in sequence. Indeed, one may consider the sequential encoding disclosed by the Katayama et al. reference as a single algorithm, with two parts. But in any event, there is no disclosure in the Katayama et al. reference of the "applying [of] different ECC algorithms to different blocks of a flash memory disk",[47] as asserted by the Examiner against claim 1, and by extension against claim 11.[48]

Therefore, Appellants submit that the Katayama et al. reference does not teach what the Examiner asserted it teaches in making the final rejection. And therefore, the teachings of the Katayama et al. reference do not "teach that the motivation of using different ECC algorithms to different blocks is to save memory space", nor do the teachings of the Katayama et al. reference "demonstrate" that the method disclosed by Bassett et al. is equally applicable to a flash memory.[49] The stated basis of the §103 rejection of claim 11 and each of its dependent claims is therefore in error.

Furthermore, there is no other stated basis by the Examiner by which one skilled in the art would be motivated to applying the teachings of the Bassett et al. reference in a flash memory. Therefore, the §103 rejection of claim 11 and its dependent claims is not based on a valid articulated reasoning with some rational underpinning, as required under current law.[50] Appellants therefore submit that the final rejection of claim 11 and its dependent claims 12 through 22 is in error, because a *prima facie* determination of obviousness of those claims has not been presented.

For these reasons, Appellants submit that the final rejection under §103 of claim 11 and its dependent claims 12 through 22 is in error, and should be reversed.

---

[47] Office Action, *supra*, pages 11, 15.
[48] Nor do the other references applied against others of the claims, nor were they asserted as doing so.
[49] Office Action, *supra*, page 12.
[50] *KSR, supra.*

Claim 23 and its dependent claims

Appellants respectfully submit that the Examiner has failed to establish a valid *prima facie* case of obviousness relative to claim 23 and its dependent claims, because the motivation and suggestion to apply the teachings of the Bassett et al. reference to a flash memory, as required to reach the system of independent claim 23 is in error as based on a misinterpretation of the Katayama et al. reference from which the Examiner found such suggestion and motivation. Furthermore, because the rejection of claim 23 is not otherwise supported, except upon this erroneous basis, Appellants submit that the final rejection of claim 23 and its dependent claims under §103 is in error and should be reversed.

As discussed above relative to claims 1 and 11, the Examiner asserted that the Bassett et al. reference taught the elements of the claim in connection with a disk drive memory, and that the Katayama et al. reference demonstrates that the Bassett et al. teachings can be also applied to a flash memory, thus reaching claim 23.[51] This assertion is based on the Examiner's interpretation of the Katayama et al. reference as teaching the applying of different ECC algorithms to different blocks of a flash memory disk.[52] However, as discussed above relative to claims 1 and 11, Appellants submit that this interpretation of Katayama et al. reference is in error, and that the reference does not, in fact, teach the applying of different ECC algorithms to different blocks of flash memory, much less means for performing error correction code calculations in the manner required by claim 23.

Rather, the Katayama et al. reference teaches the applying of two different ECC polynomials to the same block of data in sequence, but does not teach the applying of two different ECC polynomials to different flash memory blocks. As discussed above in detail relative to claims 1 and 11, the Katayama et al. reference teaches an ECC circuit 107 that derives an "outer" code check symbol C1 to a data word,[53] and an error correction coding circuit 609 that generates an "inner" check symbol C2 over a data block including the original data and the

[51] Office Action, *supra*, page 15.
[52] Office Action, *supra*, pages 11 and 12 ("Further, Katayama et al. disclose in their invention . . . a method of applying different ECC algorithms to different blocks of a flash memory disk."
[53] *Id.*, at column 11, lines 19 through 45.

"outer" code check symbol C1.[54] Each flash memory block of the Katayama et al. reference receives data encoded according to the same two algorithms, in sequence. As a result there is no disclosure in the Katayama et al. reference of the "applying [of] different ECC algorithms to different blocks of a flash memory disk",[55] as asserted by the Examiner against claim 1, and by extension against claim 23.[56]

Therefore, Appellants submit that the Katayama et al. reference does not support the basis of the §103 rejection as stated by the Examiner, because that basis is founded on a misinterpretation of the reference. Specifically, the Katayama et al. reference does not "teach that the motivation of using different ECC algorithms to different blocks is to save memory space", nor do the teachings of the Katayama et al. reference "demonstrate" that the method disclosed by Bassett et al. is equally applicable to a flash memory.[57] The stated basis of the §103 rejection of claim 23 and each of its dependent claims is therefore in error.

And because there is no other basis stated by the Examiner by which one skilled in the art would be motivated to applying the teachings of the Bassett et al. reference in a flash memory, other than this erroneous basis, the §103 rejection of claim 23 and its dependent claims is not based on a valid articulated reasoning with some rational underpinning, as required under current law.[58] Appellants therefore submit that the final rejection of claim 23 and its dependent claims 12 through 22 is in error.

For these reasons, Appellants submit that the final rejection under §103 of claim 23 and its dependent claims 24, 25, and 27 through 31 is in error, and should be reversed.

[54] *Id.*, column 11, line 46 through column 12, line 7.
[55] Office Action, *supra*, pages 11, 15.
[56] Nor do the other references applied against others of the claims, nor were they asserted as doing so.
[57] Office Action, *supra*, page 12.
[58] *KSR, supra.*

In conclusion

For the foregoing reasons, therefore, Appellants respectfully submit that the final rejection under §103 of claims 1 through 25, and 27 through 31 is in error. Reversal of the final rejection of the claims in this case is therefore respectfully requested.

Respectfully submitted,
/Rodney M. Anderson/
Rodney M. Anderson
Registry No. 31,939
Attorney for Appellants

Anderson, Levine & Lintel, L.L.P.
14785 Preston Road, Suite 650
Dallas, Texas 75254
(972) 664-9554

Claims appendix:

1. A method for encoding data associated with a page within a non-volatile memory of a memory system, the page having a data area and an overhead area, the method comprising:

dividing at least a part of the page into at least two segments of the data, the at least two segments of the data including a first segment and a second segment;

performing error correction code (ECC) calculations on the first segment according to a first ECC algorithm to encode the first segment; and

performing error correction code (ECC) calculations on the second segment according to a second ECC algorithm to encode the second segment, wherein the second segment is encoded substantially separately from the first segment.

2. The method of claim 1 wherein the first segment includes the data area and the second segment includes the overhead area.

3. The method of claim 1 wherein the first segment includes a first section of the data area and the second segment includes a second section of the data area.

4. The method of claim 1 wherein the first ECC algorithm is arranged to detect up to two incorrect bits and to correct up to one of the incorrect bits in each of the first segment and the second segment.

5. The method of claim 4 wherein the first ECC algorithm is a Hamming Code ECC algorithm.

6. The method of claim 1 wherein dividing the at least part of the page into the at least two segments of the data includes:

dividing the page into three segments, the three segments including the first segment, the second segment, and a third segment.

7. The method of claim 6 further including:

performing the ECC calculations on the third segment according to one of the first and second ECC algorithms to encode the third segment, wherein the third segment is encoded substantially separately from the first segment and the second segment.

8. The method of claim 6 wherein the first segment includes a first section of the data area, the third segment includes a second section of the data area, and the second segment includes the overhead area.

9. The method of claim 6 wherein the first segment includes a first section of the data area, the second segment includes a second section of the data area, and the third segment includes a third section of the data area.

10. The method of claim 1 wherein the non-volatile memory is one of a NAND flash memory and an MLC NAND flash memory.

11. A memory system comprising:

a non-volatile memory, the non-volatile memory including a page, the page having a data area and an overhead area, the data area and the overhead area being arranged to contain bits of data;

code devices for dividing at least a part of the page into at least two segments, the at least two segments including a first segment and a second segment;

code devices for performing error correction code (ECC) calculations according to a first ECC algorithm on the first segment to encode the first segment and on the second segment according to a second ECC algorithm to encode the second segment, wherein the second segment is encoded substantially separately from the first segment; and

a memory area for storing the code devices.

12. The memory system of claim 11 further including: a controller, the controller being arranged to process the code devices.

13. The memory system of claim 11 wherein the first segment includes the data area and the second segment includes the overhead area.

14. The memory system of claim 11 wherein the first segment includes a first section of the data area and the second segment includes a second section of the data area.

15. The memory system of claim 11 wherein the first ECC algorithm is arranged to detect up to two incorrect bits and to correct up to one of the incorrect bits in each of the first segment and the second segment.

16. The memory system of claim 15 wherein the first ECC algorithm is a Hamming Code ECC algorithm.

17. The memory system of claim 11 wherein the code devices for dividing the at least part of the page into the at least two segments include:

code devices for dividing the page into three segments, the three segments including the first segment, the second segment, and a third segment.

18. The memory system of claim 17 further including:

code devices for performing the ECC calculations on the third segment according to one of the first and second ECC algorithms to encode the third segment, wherein the third segment is encoded substantially separately from the first segment and the second segment.

19. The memory system of claim 17 wherein the first segment includes a first section of the data area, the third segment includes a second section of the data area, and the second segment includes the overhead area.

20. The memory system of claim 17 wherein the first segment includes a first section of the data area, the second segment includes a second section of the data area, and the third segment includes a third section of the data area.

21. The memory system of claim 11 wherein the non-volatile memory is one of a NAND flash memory and an MLC NAND flash memory.

22. The memory system of claim 11 wherein the code devices are one of software code devices and firmware code devices.

23. A memory system comprising:

a non-volatile memory, the non-volatile memory including a page, the page having a data area and an overhead area, the data area and the overhead area being arranged to contain bits of data;

means that divide at least a part of the page into at least two segments, the at least two segments including a first segment and a second segment; and

means that perform error correction code (ECC) calculations according to a first ECC algorithm on the first segment to encode the first segment and on the second segment according to a second ECC algorithm to encode the second segment,

wherein the second segment is encoded substantially separately from the first segment.

24. The memory system of claim 23 wherein the first segment includes the data area and the second segment includes the overhead area.

25. The memory system of claim 23 wherein the first segment includes a first section of the data area and the second segment includes a second section of the data area.

27. The memory system of claim 23 wherein the means that divide the at least part of the page into the at least two segments include:

means that divide the page into three segments, the three segments including the first segment, the second segment, and a third segment.

28. The memory system of claim 27 further including:

means that perform the ECC calculations according to one of the first and second ECC algorithms on the third segment to encode the third segment,

wherein the third segment is encoded substantially separately from the first segment and the second segment.

29. The memory system of claim 27 wherein the first segment includes a first section of the data area, the third segment includes a second section of the data area, and the second segment includes the overhead area.

30. The memory system of claim 27 wherein the first segment includes a first section of the data area, the second segment includes a second section of the data area, and the third segment includes a third section of the data area.

31. The memory system of claim 23 wherein the non-volatile memory is one of a NAND flash memory and an MLC NAND flash memory.

Evidence appendix:

None.

Related proceedings appendix:

None.